UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/796,785	03/08/2004	Albert Gordon Smith	58083-375010 (M074)	1669

72058 7590 03/22/2010
Kilpatrick Stockton LLP- Adobe Systems, Inc. 58083
Kilpatrick Stockton LLP
1100 Peachtree Street
Atlanta, GA 30309-4530

EXAMINER	
ZAHR, ASHRAF A	

ART UNIT	PAPER NUMBER
2175	

MAIL DATE	DELIVERY MODE
03/22/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/796,785	SMITH ET AL.	
	Examiner	Art Unit	
	ASHRAF ZAHR	2175	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 December 2009*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *9-19,28 and 30-34* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *9-19,28 and 30-34* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)	4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)	5) ☐ Notice of Informal Patent Application
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.	6) ☐ Other: _____.

DETAILED ACTION

1. This is the non-final action for application 10/796785. Claims 9-19, 28, 30-34 are pending in this application.

Response to Arguments

2. Applicant's arguments with respect to claims 9-19, 28, 30-34 have been considered but are moot in view of the new ground(s) of rejection.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims 9-19, 28, 30-34 are rejected under 35 U.S.C. 102(e) as being anticipated by Moser, US 2006/0200535 (Hereinafter, Moser).

Regarding Claim 9, Moser also discloses, "a method comprising: creating a root application node of a descriptor tree, responsive to a user initiating a rich internet application (RIA) defined using procedural code and declarative code". Specifically, for visualizing original model 200-T1 server-renderer 101-2 generates browser-compatible code, such as HTML, XHTML or WML. Initially, server-renderer 101-2 generates a component descriptor from original model 200-T1 and sends the component-descriptor to client 900 (Moser, ¶0083).

Moser also discloses, “generating a plurality of descriptor nodes for said descriptor tree, wherein each of said plurality describes an interface element currently instantiated and visible to said user on a currently visible pane of said RIA”. The browser compatible code contains component descriptors of the tree (Moser, ¶0095).

Moser also discloses, “responsive to said user navigating to a subsequent pane of said RIA, constructing a plurality of stacked descriptor nodes for said descriptor tree wherein each of said plurality of stacked descriptor nodes describes said interface element not instantiated and invisible to said user on said currently visible pane of said RIA and associated with said subsequent pane”. Specifically, a second example to retrieve browser increment 300-I is by using a hidden HTML iFrame element (Moser, ¶110)

Moser also discloses, “and creating a detail object from each one of: said plurality of descriptor nodes; and said plurality of stacked descriptor nodes, and rendering said interface element using a corresponding detail object”. Specifically, the result of the rendering 801 is a description of the model that can be used for its visualization. This description is then sent to client 900 (Moser, ¶135)

Regarding Claim 10, Moser also discloses, “the method of claim 9 wherein said generating comprises: generating one of said plurality of descriptor nodes for a container of said interface element not instantiated and invisible to said user on said currently visible pane of said RIA". Specifically, a second example to retrieve browser increment 300-I is by using a hidden HTML iFrame element (Moser, ¶110)

Regarding Claim 11, Moser also discloses, "the method of claim 9 wherein said association between said subsequent pane and said plurality of stacked descriptor nodes **comprises one of**: a direct link; an ordinal relationship; a statistical relationship; and a positional relationship". Specifically, the request URL is assigned to the SRC attribute of the hidden iFrame (Moser, ¶0122).

Regarding Claim 12, Moser also discloses, "a computer program product having a computer readable medium with computer program logic recorded thereon, said computer program product comprising: code for initializing of a rich internet application (RIA), wherein initializing comprises instantiating at least one visible element in said RIA". Specifically, for visualizing original model 200-T1 server-renderer 101-2 generates browser-compatible code, such as HTML, XHTML or WML. Initially, server-renderer 101-2 generates a component descriptor from original model 200-T1 and sends the component-descriptor to client 900 (Moser, ¶0083).

Moser also discloses, "and deferring instantiation of unseen ones of a plurality of stacked elements in said RIA". Specifically, a second example to retrieve browser increment 300-I is by using a hidden HTML iFrame element (Moser, ¶110)

Moser also discloses, "code for generating a descriptor tree having a plurality of descriptor nodes, wherein each of said plurality of descriptor nodes describes a plurality of instantiated and visible interface elements of said RIA". The browser compatible code contains component descriptors of the tree (Moser, ¶0095).

Moser also discloses, "code for creating one or more stacked descriptor nodes in said descriptor tree describing said unseen ones of said plurality of stacked interface elements responsive to a user navigating to said unseen ones, wherein said unseen ones are not instantiated at said starting". Specifically, in case the user interaction requires server 901 to be involved, CPP 100 sends 702 a corresponding request to CPP 101. The request makes server-controller 101-1 to modify 703 the corresponding model 200-Tn accordingly. Server-renderer 101-2 renders/generates 801 model 200-Tn after having been modified (Moser, ¶0135).

Moser also discloses, "code for rendering said plurality of visible interface elements and said unseen ones using corresponding ones of: said plurality of descriptor nodes; and said one or more stacked descriptor nodes". Specifically, The result of the rendering 801 is a description of the model that can be used for its visualization. This description is then sent to client 900 (Moser, ¶135).

Regarding Claim 13, Moser also discloses, "the computer program product of claim 12 further comprising: code for converting said plurality of descriptor nodes into a plurality of detail objects" Specifically, the result of the rendering 801 is a description of the model that can be used for its visualization. This description is then sent to client 900 (Moser, ¶135)

Moser also discloses, "code for converting said one or more stacked descriptor nodes into one or more stacked detail objects, wherein said plurality of instantiated and visible interface elements and said unseen ones are rendered directly using said plurality of detail objects and said one or more stacked detail objects". Specifically, the result of the rendering 801 is a description of the model that can be used for its visualization. This description is then sent to client 900 (Moser, ¶135)

Regarding Claim 14, Moser also discloses, "the computer program product of claim 12 wherein each one of said plurality of descriptor nodes and said one or more stacked descriptor nodes contains a software method for generating each its child nodes". Specifically, a first example to retrieve browser-increment 300-I from server 901 is by using a script tag (Moser, ¶110).

Regarding Claim 15, Moser also discloses, "the computer program product of claim 12 Further comprising: code for downloading bytecode representing said RIA to a computer of said user responsive to said initializing of said RIA, wherein said code for generating and said code for creating use said bytecode". Specifically, a second example to retrieve browser increment 300-I is by using a hidden HTML iFrame element (Moser, ¶110)

Regarding Claim 16, Moser also discloses “the computer program product of claim 13, further comprising: code for storing as a plurality of stored nodes each of: said plurality of descriptor nodes; said one or more stacked descriptor nodes; said plurality of detail objects; and said one or more stacked detail objects”. Specifically, the following coding block is a simplified HTML code example of a component descriptor of a tree component (Moser, ¶95).

Moser also discloses “code for re-rendering each of said plurality of instantiated and visible interface elements and said one or more stacked interface elements from said plurality of stored nodes.” Specifically, in case the user interaction requires server 901 to be involved, CPP 100 sends 702 a corresponding request to CPP 101. The request makes server-controller 101-1 to modify 703 the corresponding model 200-Tn accordingly. Server-renderer 101-2 renders/generates 801 model 200-Tn after having been modified (Moser, ¶0135).

Regarding Claim 17, Moser also discloses “the computer program product of claim 12 wherein said one or more stacked descriptor nodes created has a navigational relationship with a particular one of said one or more stacked interface elements to which said user navigates”. Specifically, unction increment( ) can be called when a user interaction (e.g., when the user indicates to expand second level node SL2) that affects the model of the application component (e.g., original model 200-T1) requires a change in at least one part of the corresponding browser component for its visualization.

Function increment( ) has the task to generate 420 a corresponding browser-increment (e.g., browser-increment 300-I), and write it to the output stream (Moser, ¶99).

Regarding Claim 18, Moser also discloses "the computer program product of claim 17 wherein said navigational relationship comprises one or more of: a direct link; an ordinal relationship; a statistical relationship; and a positional relationship". Specifically, the request URL is assigned to the SRC attribute of the hidden iFrame (Moser, ¶0122).

Regarding Claim 19, Moser also discloses "the computer program product of claim 12 further comprising: code for creating select ones of said one or more stacked descriptor nodes in said descriptor tree responsive to starting said RIA". Specifically, Initially, server-renderer 101-2 generates a component descriptor from original model 200-T1 and sends the component-descriptor to client 900 (Moser, ¶0083).

Regarding Claim 28, Moser also discloses, "a computer program product having a computer readable medium with computer program code recorded thereon, said computer program product comprising: program code for accessing executable code of a rich internet application, the executable code comprising code for instantiating a plurality of objects, each object for rendering a corresponding interface element of the rich internet application". Specifically, for visualizing original model 200-T1 server-renderer 101-2 generates browser-compatible code, such as HTML, XHTML or WML.

Initially, server-renderer 101-2 generates a component descriptor from original model 200-T1 and sends the component-descriptor to client 900 (Moser, ¶0083).

Moser also discloses, "program code for identifying a subset of the plurality of objects in the executable code, the subset comprising fewer than all of the plurality of objects". Specifically, for visualizing original model 200-T1 server-renderer 101-2 generates browser-compatible code, such as HTML, XHTML or WML. Initially, server-renderer 101-2 generates a component descriptor from original model 200-T1 and sends the component-descriptor to client 900 (Moser, ¶0083).

Moser also discloses, "the computer program product set forth in claim 28, further comprising program code for creating a descriptor tree comprising a plurality of descriptor nodes, each node identifying an object or container of the application". The browser compatible code contains component descriptors of the tree (Moser, ¶0095).

Moser also discloses, "program code for instantiating the objects in the subset the subset comprising fewer than of the plurality of objects". Specifically, In FIG. 2A, at T1, the user gets prompted with tree 955-1, wherein the states of the various nodes are: [0068] root-node R1: expanded (-); child nodes FL1, FL2; [0069] first level node FL1: expanded (-); child nodes: SL1, SL2; [0070] first level node FL2: collapsed; and [0071] second level nodes SL1, SL2: collapsed (+) (Moser, ¶0067).

Moser also discloses "wherein identifying the subset comprises (i) using the descriptor tree to identify a node whose object is for rendering an interface element that is not visible in the initial view and (ii) excluding the object from the subset of objects". Specifically, In FIG. 2A, at T1, the user gets prompted with tree 955-1, wherein the states of the various nodes are: [0068] root-node R1: expanded (-); child nodes FL1, FL2; [0069] first level node FL1: expanded (-); child nodes: SL1, SL2; [0070] first level node FL2: collapsed; and [0071] second level nodes SL1, SL2: collapsed (+).(Moser, ¶0067).

Moser also discloses "program code for rendering an initial view of the application using the instantiated objects". Specifically, FIG. 2D, at T4, shows an example of the initial state of tree 955-1. Root-node R1 has status collapsed (+) and none of its child nodes is visualized (Moser, ¶0075).

Moser also discloses "program code for instantiating at least one other object of the plurality of objects in response to user interaction with an interface element of the initial view". Specifically, a second example to retrieve browser increment 300-I is by using a hidden HTML iFrame element (Moser, ¶110)

Moser also discloses "program code for rendering another view of the application using the instantiated at least one other object". Specifically, the result of the rendering 801 is a description of the model that can be used for its visualization. This description is then sent to client 900 (Moser, ¶135)

Regarding Claim 30, Moser also discloses "the computer program product set forth in claim 28, wherein using the descriptor tree to identify a node whose object is for rendering an interface element that is not visible in the initial view comprises<u>: determining if the node identifies itself as corresponding to a stacked navigation object".</u>

Specifically, a hidden HTML iFrame element is excluded unless retrieved through a browser increment (Moser, ¶110)

Regarding Claim 31, Moser also discloses "the computer program product set forth in claim 28, wherein creating a descriptor tree comprising <u>identifying a node whose object is for rendering an interface element that is not visible in the initial view as a hidden node</u>"; and "<u>wherein objects associated with hidden nodes are excluded from the subset</u>". Specifically, a hidden HTML iFrame element is excluded unless retrieved through a browser increment (Moser, ¶110)

Regarding Claim 32, Moser also discloses "A method comprising: accessing code of a rich internet application from a computer-readable medium; generating a descriptor tree based on the accessed code, the descriptor tree having a plurality of descriptor nodes corresponding to respective elements of the rich internet application". Specifically, the browser compatible code contains component descriptors of the tree (Moser, ¶0095).

Moser also discloses "at least some of the descriptor nodes identifying interface elements of the rich internet application intended to be instantiated and visible at a beginning of the rich interact application" Specifically, initially, server-renderer 101-2 generates a component descriptor from original model 200-T1 and sends the component-descriptor to client 900 (Moser, ¶0083).

Moser also discloses "creating at least one hidden descriptor node in the descriptor tree, the hidden node identifying an interface element not intended to be instantiated or visible at the beginning of the rich internet application". Specifically, FIG. 2D, at T4, shows an example of the initial state of tree 955-1. Root-node R1 has status collapsed (+) and none of its child nodes is visualized (Moser, ¶0075).

Moser also discloses "instantiating and rendering a plurality of interface elements at the beginning of the rich internet application based on corresponding non-hidden nodes in the descriptor tree". Specifically, initially, server-renderer 101-2 generates a component descriptor from original model 200-T1 and sends the component-descriptor to client 900 (Moser, ¶0083).

Moser also discloses "responsive to data identifying navigation in the rich internet application, instantiating and rendering a second interface element based on a corresponding hidden descriptor node". Specifically, a second example to retrieve browser increment 300-I is by using a hidden HTML iFrame element (Moser, ¶110)

Moser also discloses "wherein instantiation of the second interface element is deferred until the navigation occurs". Specifically, in case the user interaction requires server 901 to be involved, CPP 100 sends 702 a corresponding request to CPP 101. The request makes server-controller 101-1 to modify 703 the corresponding model 200-Tn accordingly. Server-renderer 101-2 renders/generates 801 model 200-Tn after having been modified (Moser, ¶0135).

Regarding Claim 33, Moser also discloses "the method set forth in claim 32, wherein each of the descriptor nodes contain a software method for generating each of its child nodes, if any". Specifically, a first example to retrieve browser-increment 300-I from server 901 is by using a script tag (Moser, ¶110).

Regarding Claim 34, Moser also discloses "the method set forth in claim 32, further comprising prior to accessing the executable code, downloading the code." Specifically, the browser increment is sent to the client controller (Moser, ¶0124).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to ASHRAF ZAHR whose telephone number is (571)270-1973. The examiner can normally be reached on M-F 9:30 am - 6 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Bashore can be reached on (571)272-4088. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Ashraf Zahr/
Examiner, Art Unit 2175